

CLAIMS

What is claimed is:

1. A method for iteratively traversing a hierarchical circuit design, comprising:
 - selecting an initial net and an instance history that uniquely defines the initial net within the circuit design;
 - appending, to a list of nets to be processed, the initial net and the instance history;
 - inserting, into a set of visited nets, the initial net and the instance history;
 - visiting, in response to a first request from a user, each additional net connected to the initial net; and
 - returning, in response to a second request from the user, the initial net and each additional net.
2. The method of claim 1, the step of visiting comprising:
 - determining, for each one of the nets to be processed in the list of nets to be processed, the additional nets that are connected to the net to be processed;
 - checking the set of visited nets to determine unvisited additional nets that have not been visited;
 - appending, to the list of nets to be processed, each of the unvisited additional nets; and
 - inserting, into the set of visited nets, each of the unvisited additional nets.
3. The method of claim 2, wherein the step of determining comprises:
 - calling, for each instance in the hierarchical circuit design connected to the net to be processed, a first user-defined callback function to identify instances of interest;
 - querying, for each instance of interest, a second user-defined callback function to identify the additional nets; and
 - appending, to the list of nets to be processed, the additional nets identified by the second user-defined callback function.

4. The method of claim 3, wherein the first callback function is configured to provide an indication of the instances to be traced around.

5. The method of claim 3, wherein the first callback function is configured to provide an indication of the instances for which tracing should halt.

6. The method of claim 5, wherein the second callback function is configured to resume a trace at one or more nodes when a first node of the instance to be traced around is encountered during the trace, and to terminate the trace if the first node is a specified type of terminal on the instance for which tracing should halt.

7. The method of claim 3, wherein the instances of interest comprise a specified type of transistor.

8. A system for iteratively traversing a hierarchical circuit design, comprising:
means for selecting an initial net and an instance history that uniquely defines the initial net within the design;
means for appending, to a list of nets to be processed, the initial net and the instance history;
means for inserting, into a set of visited nets, the initial net and the instance history; and
means for visiting, in response to a first request from a user, each additional net connected to the initial net.

9. The system of claim 8, wherein the initial net and each additional net is returned in response to a second request from the user and wherein the means for visiting further comprises:

means for determining, for each one of the nets to be processed in the list of nets to be processed, the additional nets that are connected to the net to be processed;
means for checking the set of visited nets to determine unvisited additional nets that have not been visited;
means for appending, to the list of nets to be processed, each of the unvisited additional nets; and

means for inserting, into the set of visited nets, each of the unvisited additional nets.

10. The system of claim 9, wherein the means for determining comprises:
means for calling, for each instance in the design connected to the net to be processed, a first user-defined callback function to identify instances of interest;
means for querying, for each instance of interest, a second user-defined callback function to identify the additional nets; and
means for appending, to the list of nets to be processed, the additional nets identified by the second user-defined callback function.

11. The system of claim 10, wherein the first user-defined callback function is configured to allow a user to provide an indication of the instances to be traced around.

12. The system of claim 10, wherein the first user-defined callback function is configured to allow a user to provide an indication of the instances for which tracing should halt.

13. The system of claim 12, wherein the second user-defined callback function is configured to resume a trace at one or more nodes when a first node of the instance to be traced around is encountered during the trace, and to terminate the trace if the first node is a specified type of terminal on the instance for which tracing should halt.

14. A system for iteratively traversing a hierarchical circuit design, comprising:
an iterator function;
an incomplete trace object;
a processor for executing the iterator function and invoking the incomplete trace object, wherein the iterator function:
selects an initial net and an instance history that uniquely defines the initial net within the circuit design;

appends, to a list of nets to be processed, the initial net and the instance history;
inserts, into a set of visited nets, the initial net and the instance history;
and wherein the incomplete trace object:
visits, in response to a first request from a user, each additional net connected to the initial net; and
returns, in response to a second request from the user, the initial net and each additional net.

15. The system of claim 14, wherein each additional net is visited by:
determining, for each one of the nets to be processed in the list of nets to be processed, the additional nets that are connected to the net to be processed;
checking the set of visited nets to determine unvisited additional nets that have not been visited;
appending, to the list of nets to be processed, each of the unvisited additional nets; and
inserting, into the set of visited nets, each of the unvisited additional nets.

16. The system of claim 15, including a first user-defined callback function and a second user-defined callback function, wherein:
the first user-defined callback function is called to identify instances of interest for each instance in the circuit design connected to the net to be processed; and
the second user-defined callback function is queried to identify the additional nets for each instance of interest, and to append the additional nets thus identified to the list of nets to be processed.

17. The system of claim 16, wherein the first user-defined callback function is configured to allow a user to provide an indication of the instances to be traced around, or for which tracing should halt.

18. The system of claim 17, wherein the second user-defined callback function is configured to resume a trace at one or more nodes when a first node of the

instance to be traced around is encountered during the trace, and to terminate the trace if the first node is a specified type of terminal on the instance for which tracing should halt.

19. A software product comprising instructions, stored on computer-readable media, wherein the instructions, when executed by a computer, iteratively traverse a hierarchical circuit design, comprising:

- instructions for selecting an initial net and an instance history that uniquely defines the initial net within the circuit design;
- instructions for appending, to a list of nets to be processed, the initial net and the instance history;
- instructions for inserting, into a set of visited nets, the initial net and the instance history;
- instructions for visiting, on a first request from a user, each additional net connected to the initial net; and
- instructions for returning, on a second request from the user, the initial net and each additional net.

20. The software product of claim 19, wherein the instructions for visiting further comprise:

- instructions for determining, for each one of the nets to be processed in the list of nets to be processed, the additional nets that are connected to the net to be processed;
- instructions for checking the set of visited nets to determine unvisited additional nets that have not been visited;
- instructions for appending, to the list of nets to be processed, each of the unvisited additional nets; and
- instructions for inserting, into the set of visited nets, each of the unvisited additional nets.